We Claim:

1. A sizing algorithm for sizing a parent grayscale pixel map expressing an image having edges using a computer, before projection onto a sensitive recording surface wherein the pixels have a size R_p, comprising the steps of:

(i) inputting the parent grayscale pixel map image with edges where an edge is defined by gray pixels having values between 1, 2, …n, or by pixels having at least one 0-gray value (black 0-dose level) pixel neighbor;

(ii) calculating a grayscale correction value (g) equal to a sizing distance ***S*** parameterized by a machine constant equal to R_p divided by the number of grayscale values;

(iii) finding and flagging edge pixels expressed within a frame of the parent pixel map; and

(iv) finding and flagging corner edge pixels within the frame;

(v) sliding a sub-matrix window within the frame, to calculate and store gradient values for each edge pixel relative to the edges within the frame;

(vi) looping over pixels within the frame to adjust the grayscale value of edge and corner pixels and neighboring pixels;

(vii) propagating new grayscale values per the grayscale correction value (g) to pixels from each adjusted edge and corner pixel within the frame in a direction normal to each edge to establish a new edge position within the frame, and

(viii) where the parent pixel map is composed of a plurality of frames, reassembling the frames generating a daughter grayscale pixel map expressing a different size image than that expressed in the parent pixel map system which upon projection and recording, compensates for expected systemic distortions.

2. The algorithm of claim 1 where the frame is a 5X5 matrix, and the sub-matrix window is a 3X3 matrix.

3. A sizing algorithm for downsizing a parent grayscale pixel map having pixels of size R_p and grayscale values between 0, 1, 2, …n, expressing an image having edges with grayscale values ranging from 1, 2,…n, using a computer, before projection onto a sensitive recording surface comprising the steps of:

calculating a factor g equal to a desired sizing distance $\boldsymbol{S}$ divided by a machine constant K_m equal to pixel size R_p divided by the number of grayscale values;

setting

$$G'(i,j) = Max(G(i,j) - g, 0), \text{ and}$$
$$\vec{\delta}G(i,j) = \left|G(i,j) - g\right| \bullet (\nabla_x, \nabla_y)$$

looping over pixels

```
{
        if( pixel ij is an edge pixel)
        {
                ∇(i, j) = estimated gradient;
                storing the value G′( i, j) of the new pixel;
                if(‖δG(i, j)‖) > 0
                {
                propagating vector differences to neighboring pixels along
                the gradient direction;
                }
        }
}
end.
```

4. A sizing algorithm for upsizing a parent grayscale pixel map having pixels of size R_p and grayscale values between 0, 1, 2, ...n, expressing an image having edges with grayscale values ranging from 1, 2,...n, using a computer, before projection onto a sensitive recording surface comprising the steps of:

   calculating a factor g equal to a desired sizing distance $\boldsymbol{S}$ divided by a machine constant K_m equal to pixel size R_p divided by the number of grayscale values;

   setting

$$G'(i,j) = Min(G(i,j) + g, gmax), \text{ and}$$
$$\vec{\delta}G(i,j) = \left| gmax - \{G(i,j) + g\} \right| \bullet (\nabla_x, \nabla_y)$$

   looping over pixels
   {
      if(pixel ij is an edge pixel)
      {
         $\nabla(i,j) =$ estimated gradient;
         storing the value G′(i, j) of the new pixel;
         $if\left(\left\|\vec{\delta}G(i,j)\right\|\right) > 0$
         {
         propagating vector differences to neighboring pixels along the gradient direction;
         }
      }
   }
   end.

5. The sizing algorithm of claim 1, or 3 or 4 wherein edge pixels are flagged by successively mapping a sub matrix array **G** of pixel grayscale values from the parent grayscale pixel map into a an edge matrix **E** with a Boolean procedure for counting the 0-gray value pixels and for assigning a value to each pixel of 0, 1, or 2, where 0 indicates a particular pixel is not an edge pixel, 1 indicates a particular pixel is in a class consisting of inclined edge pixels and corner pixels, and 2 indicates a particular pixel is an edge pixel.

6. The sizing algorithm of claim 5 wherein the sub matrix array **G** is a 3X3 matrix.

7. The sizing algorithm of claim 6 wherein diverging corner edge pixels are flagged by successively mapping each edge matrix **E** into a Boolean **I** returning true if **E** has an edge_sum equal to 5 indicating a particular edge pixel is a diverging corner edge pixel, else returning false.

8. The sizing algorithm of claim 5 wherein new grayscale values are propagated normal to edges of the parent grayscale pixel map to new edge positions with:
   (i) a 45° rule computation operator for such edges inclined at 45° ($\pi/4$) relative to an orthogonal coordinate of the parent pixel map; and
   (ii) a non-45° rule computation operator for such edges inclined at angles other than 45° relative to an orthogonal coordinate of the parent pixel map; and
   (iii) a gray-to-neighbors computation operator for propagating gray values to pixels neighboring such edges in the direction of the gradient.